Linear Regression

Week 3

There are 3 types of Prediction Errors

1. Bias Error
2. Variance Error
3. Irreducible Error

Bias Error

Biases are assumptions made by model to make learning easier and quicker

Having high bias makes the model less flexible to change and low performance on complex tasks

Having low bias means, model assumes less things about the target function

Variance Error

The amount of change to make in order to reach the target function

It is ok for many machine learning tasks to have some amount of variance from training data.

Simpler linear algorithms mostly have low variance, but compared to those, more complex algorithms tend to have high variance

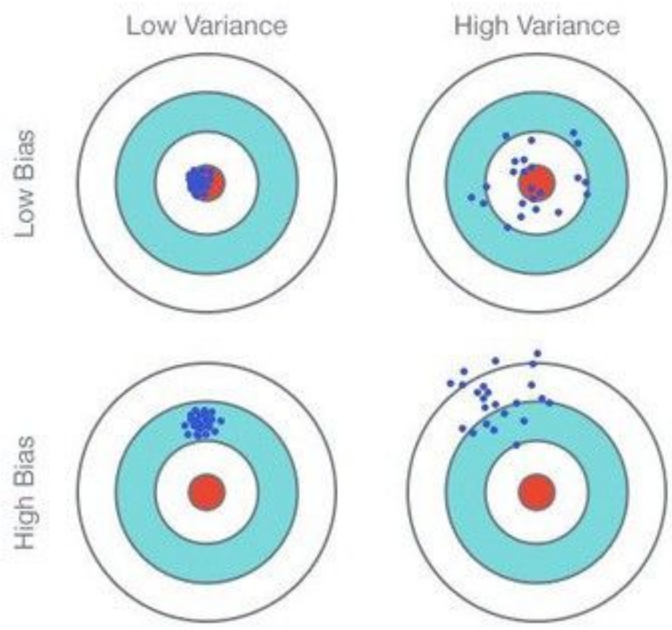


Fig. 1: Graphical Illustration of bias-variance trade-off , Source: Scott Fortmann-Roe., Understanding Bias-Variance Trade-off

Image source: https://medium.com/@mp32445/understanding-bias-variance-tradeoff-ca59a22e2a83

Bias Variance Tradeoff

The goal is to achieve low bias, and low variance to get high prediction score.

Increasing the bias will decrease the variance

Decreasing the bias will increase the variance

We should be aware of this and design the algorithms accordingly

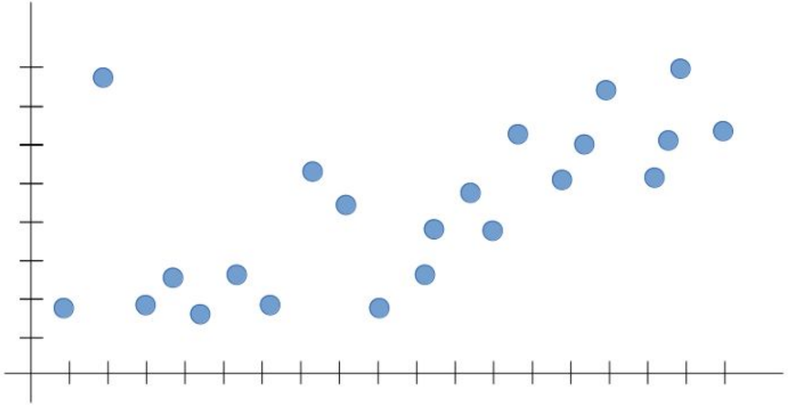
The state of the model

We want for model to be able to generalize

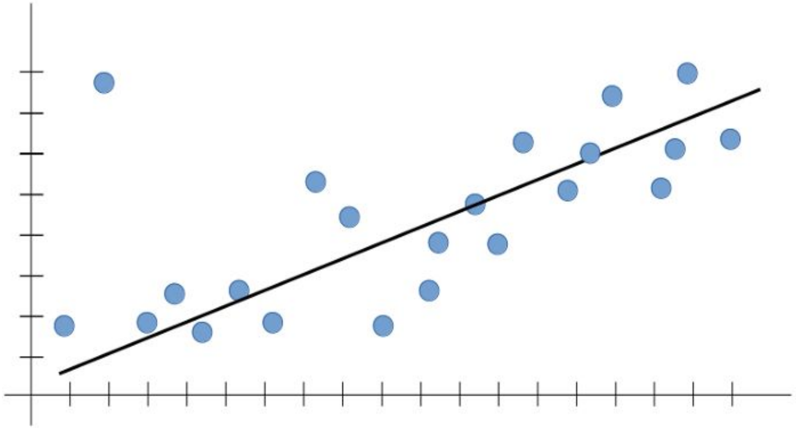
However sometimes the model starts to memorize the training data and performs worse during testing

Training time, training samples, etc. are factors causing model to perform worse during testing.

This issue is called **Overfitting (high variance).**



The Dataset



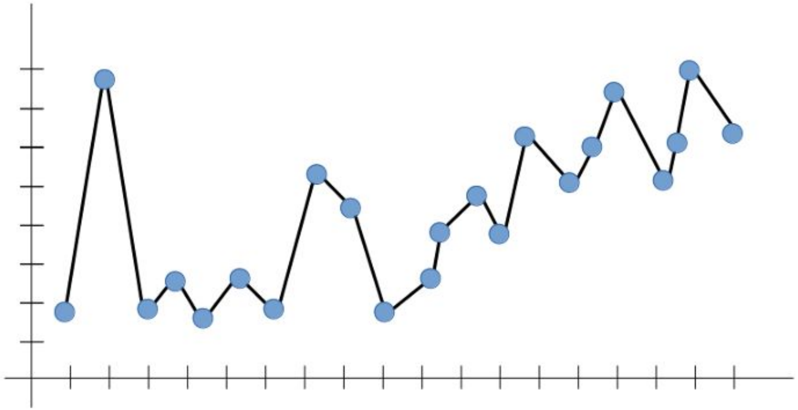


Image source: https://towardsdatascience.com/what-are-overfitting-and-underfitting-in-machine-learning-a96b30864690

To prevent overfit

Early Stopping

Regularization

Train with more data

Train with simple data

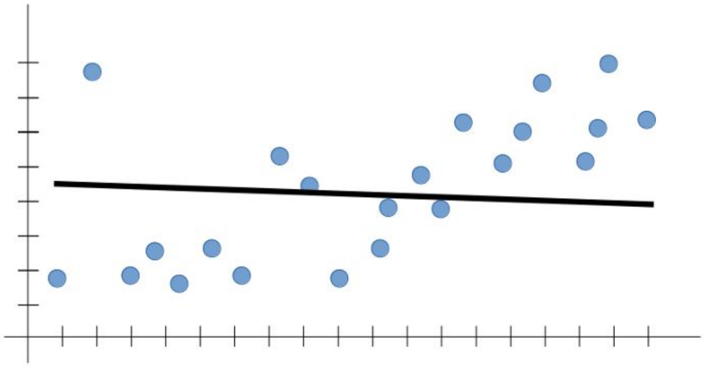
Ensembling

Underfitting

To prevent overfitting what can be done?

Early stopping - This may cause model not able to learn much in given time.

Training less can cause model to **underfit (high bias)** which will also predict poorly during testing.



Linear Regression

$$y = w_0 + w_1 x_1 + w_2 x_2 + ... w_n x_n$$

$$\begin{array}{cc} & \text{Features (n)} \\ \text{Samples (m)} & \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ & \cdots & \cdots & \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix} \end{array} \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_n \end{bmatrix} + \mathbf{b} = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{bmatrix}$$

$$X_{sample, feature}$$

Loss Function

For linear regression, MSE (mean squared error) is used to calculate the distance between the desired value ***Y*** and calculated value ***Ŷ*.**

$$J(w) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

Optimization

$$J(w) = \frac{1}{m}\sum_{i=1}^{m}((wx_i + b) - y_i)^2$$

It is time to better ourselves up

We need to change our weight to better capture the data

$$\partial W = \frac{2}{m}\sum_{i=1}^{m}((wx_i + b) - y_i)(x_i)$$

$$\partial b = \frac{2}{m}\sum_{i=1}^{m}((wx_i + b) - y_i)$$

$$w := w - \alpha * \partial W$$

$$b := b - \alpha * \partial b$$